

A Parallel-efficient GPU Package for Multiphase Flow in Realistic Nano-pore Networks

Yidong Xia

Materials Science and Engineering
Idaho National Laboratory
Idaho Falls, Idaho, USA
yidong.xia@inl.gov

Lixiang Luo

Center of Excellence at ORNL
IBM Research
Oak Ridge, Tennessee, USA
Lixiang.Luo@ibm.com

Joshua Kane

Materials Science and Engineering
Idaho National Laboratory
Idaho Falls, Idaho, USA
joshua.kane@inl.gov

Ansel Blumers

Department of Physics
Brown University
Providence, Rhode Island, USA
ansel_blumers@brown.edu

Jan Goral

Chemical Engineering
University of Utah
Salt Lake City, Utah, USA
jgoral@egi.utah.edu

Yu-Hang Tang

Computational Research Division
Lawrence Berkeley National Lab
Berkeley, California, USA
tang@lbl.gov

Zhen Li

Applied Mathematics Division
Brown University
Providence, Rhode Island, USA
zhen_li@brown.edu

Matthew Andrew

Marketing Technology
Carl Zeiss X-Ray Microscopy
Pleasanton, California, USA
matthew.andrew@zeiss.com

ABSTRACT

Simulations of fluid flow in oil/gas shale rocks are challenging in part due to the heterogeneous pore sizes ranging from a few nanometers to a few micrometers. Additionally, the complex fluid-solid interaction occurring physically and chemically must be captured with high resolution. To address these challenges while minimizing computational cost, we present a GPU code that has implemented a many-body dissipative particle dynamics (mDPD) model for multiphase flow in shale. Realistic nano- to micro-pore channels in shale are constructed from 3D high-resolution stack images. In our benchmark tests, the code delivers nearly perfect weak and strong scalings on up to 512 K20X GPUs on Oak Ridge National Laboratory (ORNL) Titan supercomputer. Moreover, single-GPU benchmarks on the DGX-1 (V100/no NVLink), ORNL's SummitDev (P100/NVLink 1.0) and Summit (V100/NVLink 2.0) suggest that the latest Host-to-Device NVLink can significantly boost overall performance, in addition to the Device-to-Device NVLink.

CCS CONCEPTS

• Hybrid/heterogeneous/accelerated algorithms • Computational fluid dynamics and mechanics • Computational design optimization for aerospace, energy, manufacturing & industrial applications

KEYWORDS

GPU, shale, nano-porous, multiphase flow

1 Background

Shales contain most of the world's oil and gas resources on Earth. However, only a small fraction of the resources, most of which are contained in kerogen-hosted pores (Fig. 1), can be recovered. Unlike conventional reservoirs in which oil and gas transport can be described by continuum models based on the Navier-Stokes (N-S) equations [1], fluid flow in nano-/micro-porous shale is strongly affected by thermal fluctuation and molecular diffusion near fluid-fluid and fluid-solid interfaces (Fig. 2). For a better understanding

of how fluids flow in shale, a mDPD based mesoscale model is employed for single- and multi-phase flow simulation in realistic shale pore geometries [2]. In order to predict critical materials properties such as permeability and relative permeability in representative 3D domains, flow simulations must be conducted in sufficiently large nano-/micro-porous systems, which may require $10^8 \sim 10^9$ particles and $10^7 \sim 10^8$ timesteps. Efficient HPC strategies such as GPUs are highly encouraged for accelerated computing.

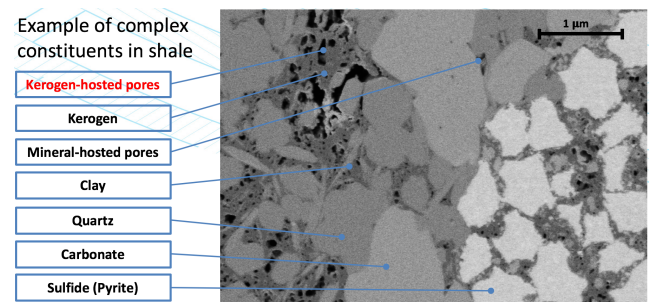


Figure 1: Example of complex constituents in a shale sample.

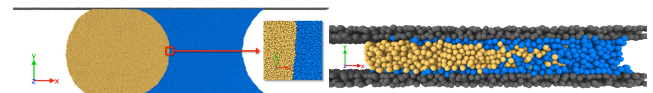


Figure 2: Comparison of fluid-fluid and fluid-solid interfaces in a 100-nm pore (left) and in a 2-nm pore (right).

2 Approach

USERMESO is a massively-parallel, hybrid CPU/GPU code for DPD particle flow and transport simulations. The code structure aligns with that of LAMMPS[3] and is thus as customizable as LAMMPS. Version 1.0 of the code was developed by Tang et. al. [4] with key features such as 1) an atomics-free warp-synchronous neighbor list construction algorithm, 2) a 2-level particle reordering scheme which aligns with the cell list lattice boundaries for generating

strictly monotonic neighbor list, 3) overlapping pairwise force evaluation with halo exchange using CUDA streams for hiding communication and kernel launch latency, and 4) radix sort with GPU stream support. Blumers et al. [5] extended the code in version 2.0 by implementing and optimizing a red blood cell transport model.

To address the computational challenges and capability gap outlined in Section 1, we present a GPU implementation of the mDPD model (with codename *USER*MESO 2.5). In mDPD[6], the conservative force $\mathbf{F}_{ij}^C$ is expressed as sum of a long-range attractive force that is density-independent, and a short-range repulsive force that depends on a weighted average of the local particle density $\bar{\rho}_i$, i.e.,

$$\mathbf{F}_{ij}^C = A_{ij}w^C(r_{ij})\hat{\mathbf{r}}_{ij} + B_{ij}(\bar{\rho}_i + \bar{\rho}_j)w_d(r_{ij})\hat{\mathbf{r}}_{ij} \quad (1)$$

Compared with the standard DPD, mDPD requires an additional data transfer between devices and hosts to update $\bar{\rho}_i$ for every time step. A no-slip boundary condition for pore channels with arbitrarily complex geometries [7] is also implemented in *USER*MESO 2.5 to prevent fluid particles from penetrating pore surfaces. Realistic nano-/micro-pore channels are constructed from 3D stack images, which are obtained through focused ion beam scanning electron microscope (FIB-SEM) with 2.5-nm unit pixel resolution. Fig. 3 illustrates a workflow for 3D digital imaging to simulations of shale oil recovery in a region of interest through pressurized flooding, where fluid-permeability characterization can be conducted.

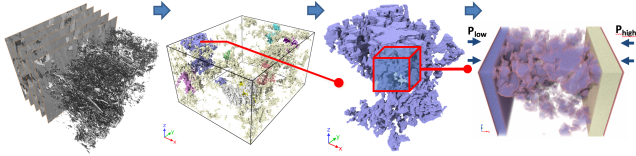


Figure 3: From 3D digital imaging to porous flow simulations.

2 Result

Preliminary benchmarks, e.g. performance tests, strong- and weak-scaling of pore-network flow were conducted using HPC resources at ORNL, INL and IBM. For simplicity, we benchmarked our code for a body force-driven flow in a domain with periodic boundaries and uniform pore distribution and. The pore network is represented by the space (narrowest gap = 4 nm) between the circles (14 nm in diameter), which symbolize the wall of the porous channel. A no-slip boundary condition is prescribed at the wall surfaces, which prevents fluid particles from penetrating the walls. Our tests show that *USER*MESO 2.5 delivers nearly perfect strong- and weak-scalings up to 512 K20X GPUs on ORNL’s Titan, shown in Fig. 4. For a direct translation of performance to modern GPUs, single-GPU benchmarks were also performed with different generations of accelerators, and the result is shown in Fig. 5. It is found that the performance is significantly boosted when CPU is connected to GPU via NVLink, as the case on ORNL’s SummitDev and Summit. Lastly, we show in Fig. 6 that the speedup brought by NVLink is evident from inspecting the breakdown of walltime of a single-GPU simulation on GPU related tasks. On DGX-1 (V100) with no NVLink, transfers take up 53% of GPU related walltime. On SummitDev with Host-to-Device NVLink1, transfers only take 24%.

When both V100 and NVLink2 are available, walltime on GPU related tasks is further shortened.

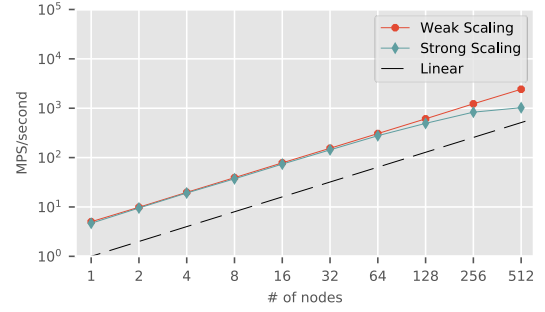


Figure 4: Strong- and weak-scaling. *USER*MESO 2.5 delivers nearly perfect scaling on up to 512 GPUs on ORNL’s Titan. For the strong-scaling, the system size is kept at 2.1 million particles; whereas, for the weak-scaling, the system size is fixed at 1 million particles per node.

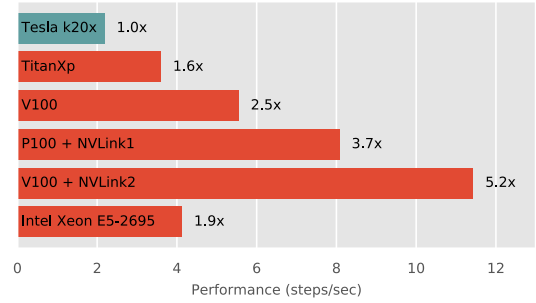


Figure 5: Single-GPU performance comparison of different generations of NVIDIA GPUs. We found that Host-to-Device NVLink can significantly boost performance, as the case on ORNL’s SummitDev and Summit.

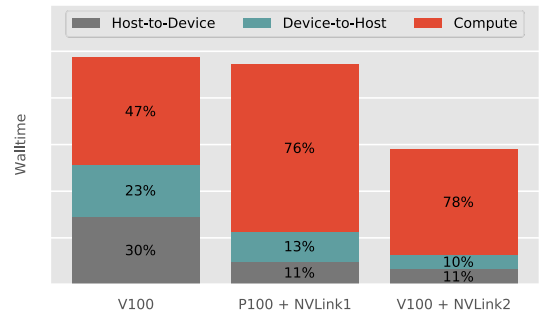


Figure 6: Breakdown of walltime of a single-GPU simulation on GPU related tasks for DGX-1, SummitDev and Summit.

ACKNOWLEDGMENTS

This work is supported by the Idaho National Laboratory (INL) Laboratory Directed Research & Development (LDRD) Program under Department of Energy Idaho Operations Office Contract DE-AC07-05ID14517. This work also used resources at Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-AC05-00OR22725.

REFERENCES

- [1] Bear, J. (1972). Dynamics of Fluids in Porous Media. Dover Publications, Inc., New York, USA.
- [2] Xia, Y., Goral, J., Huang, H., Miskovic, I., Meakin, P., & Deo, M. (2017). Many-body dissipative particle dynamics modeling of fluid flow in fine-grained nanoporous shales. *Physics of Fluids*, 29(5), 056601.
- [3] Plimpton, S. (1995). Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117(1), 1-19.
- [4] Tang, Y. H., & Karniadakis, G. E. (2014). Accelerating dissipative particle dynamics simulations on GPUs: Algorithms, numerics and applications. *Computer Physics Communications*, 185(11), 2809-2822.
- [5] Blumers, A. L., Tang, Y. H., Li, Z., Li, X., & Karniadakis, G. E. (2017). GPU-accelerated red blood cells simulations with transport dissipative particle dynamics. *Computer Physics Communications*, 217, 171-179.
- [6] Warren, P. B. (2003). Vapor-liquid coexistence in many-body dissipative particle dynamics. *Physical Review E*, 68(6), 066702.
- [7] Li, Z., Bian, X., Tang, Y. H., & Karniadakis, G. E. (2018). A dissipative particle dynamics method for arbitrarily complex geometries. *Journal of Computational Physics*, 355, 534-547..
- [8] Tang, Y. H., Lu, L., Li, H., Evangelinos, C., Grinberg, L., Sachdeva, V., & Karniadakis, G. E. (2017). OpenRBC: a fast simulator of red blood cells at protein resolution. *Biophysical Journal*, 112(10), 2030-2037.